```
**************************************************************
*                                                            *
*              U N I V E R S I T Y   O F   U T A H           *
*                                                            *
*     C O M P U T E R   S C I E N C E   D E P A R T M E N T  *
*                                                            *
**************************************************************
```

MONDAY 11/10/80 AT 04:12:03 PM

```
L       EEEE PPPP  RRRR   EEEE  AAA  U   U
L     E     P   P R   R E      A   A U   U
L     E     P   P R   R E      A   A U   U
L     EEEE  PPPP  RRRR  EEEE  AAAAA U   U
L     E     P     R R   E      A   A U   U
L     E     P     R  R  E      A   A U   U
 LLLL  EEEE P     R   R  EEEE A   A  UUU

 AAA   M M  PPPP   SSS          M M   AAA  IIIII L          TTTTT X   X TTTTT          1
A   A M M M P   P S   S         M M M A   A   I   L            T   X   X   T           11
A   A M M M P   P S             M M M A   A   I   L            T    X X    T           1
AAAAA M M M PPPP   SSS          M M M AAAAA   I   L            T     X     T           1
A   A M   M P          S        M   M A   A   I   L            T    X X    T           1
A   A M   M P      S   S        M   M A   A   I   L       ..   T   X   X   T     ..    1
A   A M   M P       SSS         M   M A   A IIIII LLLL    ..   T   X   X   T     ..   111
```

7-Jul-80 14:54:20-MDT,0000971;000000000003

Date: 29 Jun 1980 1546-MDT
From: KELLER
Subject: FGL use
To: AMPS:
Remailed-date: 7 Jul 1980 1454-MDT
Remailed-from: BENSON
Remailed-to: lepreau

A new version of the FGL compiler eliminates a few current annoyances.
Pseudo-functions which cause side-effects are NO LONGER identified in
common sub-expressions. This means that the ugly requirement of adding
an extra argument is no longer there. The exact list of functions for
which this is true can be found by examination of <amps>fgldef.red under
FLAG('...., FGLNOSUBEXP). It is also possible to prevent identification of
any functions, except literals, by using the keyword PROCEDURE instead of
FUNCTION. It would be a good idea to use PROCEDURE only for pseudo-
functions which cause side-effects and (conversely) FUNCTION only
for pure functions. A few other niceties will be reported in the forthcoming
revision of the FGL PROGRAMMERS' GUIDE. Watch for it on your newstands.

10-Jul-80 14:00:15-MDT,0000001082;000000000003
Date: 10 Jul 1980 1400-MDT
From: KELLER
Subject: Applicative programming workshop
To: AMPS:

At the workshop on applicative languages earlier this week, it was pointed
out, and conceded by Friedman and Wise, that

```
frons(x,y) = merge(cons(x, 'nil), y),
```

thereby rendering a concept formerly explained in a most baroque manner
fairly easy to understand (see fgl code for merge in <amps>merge.fgl).

Doubt was also cast on another try at giving a denotational semantics for
merge, which was presented by well-known computer-scientist/logician David
Park, who has been working on the problem for 8 years.

Another baroque explanation, this time of I-structures (Arvind and Thomas),
seems to be done much more easily using tuples/make, at least for the
integer index case, which is what they are proposing to implement.

The list of speakers also included Ackermann, Ashcroft, Buneman, Frankel, Grit,
Hewitt, Jayaraman, Keller, Lindstrom, MacQueen, Mago, Plouffe, Sleep, Smoliar.
Information on any of these presentations can be obtained by asking me.

13-Jul-80 03:02:23-MDT,00001307;000000000003
Date: 13 Jul 1980 0302-MDT
From: KELLER
Subject: fgl compiler
To: AMPS:

The present fgl compiler has a few new features over its predecessor,
including

a) Tail recursion is automatically introduced. You don't need to
include the -s anymore, unless you need to optimize where

the compiler can't. (Eventually ¬s will become illegal.)

b) A function eol() generates an "end-of-line" character, so that print routines can do all the formatting externally, allowing a totally applicative style of programming with a universal print routine.

b) Expressions may be used as the function to be applied in an application. The operand must be parenthesized. Thus one could write (if p then f else g)(if q then a else b) instead of if p then f(if q then a else b) else g(if q then a else b) when f and g are function arguments. This also allows us to do multi-d arrays, since applications of this form are LEFT associative. See <amps>3d.fgl for an example. Note that function compositions involving single identifiers take precedence and are still right associative. That is f g h a is f(g(h(a))).

Let me know if you have any problems (if so, use the older version of fgl.exe in the meantime). Thanks.

17-Jul-80 12:15:25-MDT,0000000227;000000000003
Date: 17 Jul 1980 1215-MDT
From: KELLER
Subject: fgl documentation
To: AMPS:
cc: hearn, organick

A new version of the FGL Programmers' guide is now in <amps>fgl.doc.
Hard copies are available from Keller.

28-Jul-80 13:45:26-MDT,00000000536;000000000003
Date: 28 Jul 1980 1345-MDT
From: KELLER
Subject: Simulator changes
To: AMPS:

The latest version of the simulator prints the "plies" (i.e. concurrent execution slices) rather than the "simsteps" on the tty. It also includes in sim.out a breakdown of the work done intc different types of tasks. The figures are interesting. It seems that about 25% of the work is in execution of "data processing" tasks (OPNs) while most of the rest is in reference counting. Fetch/Forwards seem to take surprisingly little.

22-Sep-80 16:54:58-MDT,00000476;000000000003
Date: 22 Sep 1980 1654-MDT
From: KELLER
Subject: Change in convention
To: AMPS:

The internal flag REFQUEUE in the simulator is now defaulted to OFF. This means that reference count operations don't get simulated as tasks, so your program runs faster and costs less. It also means that your concurrency will be decreased by a factor of roughly 3. If you want

the added realism of reference counting, you may include
#ON REFQUEUE# in your FGL source.

1-Oct-80 15:31:27-MDT,0000000424;000000000003
Date: 1 Oct 1980 1531-MDT
From: LINDSTROM
Subject: new paper
To: AMPS: ;

There is a draft of a new paper available, entitled "Parallelism in Functional Programming through Applicative Loops", by Keller & Lindstrom. It is largely, but not exclusively, an updated version of the earlier paper "Programming Paradigms in a Graph-Based Lisp Extension". Comments are invited; ask for a copy from Bob or me.

9-Oct-80 09:54:37-MDT,0000001046;000000000003
Date: 9 Oct 1980 0954-MDT
From: KELLER
Subject: [KELLER: Multiprocessing Systems]
To: AMPS:

Date: 9 Oct 1980 0952-MDT
From: KELLER
Subject: Multiprocessing Systems
To: CS-Graduate-Students:
cc: bboard

The AMPS (Applicative Multiprocessing Systems) Group will begin its regular weekly meetings this Friday, 11-12 in 3141. The first meeting will hear brief "banner" presentations on work by the following graduate students: Rima Doany, Paul Hudak, Francie Hunt, Barat Jayaraman, Chu-Shik Jhon, Krithi Ramamritham, and Jonathan Yen. Additionally, new software developments will be mentioned, and plans for the rest of the quarter's meetings will be discussed.

It is suggested that we have a new format this term, alternating current research discussions with tutorials on AMPS conception and FGL programming, including presentation of internal aspects of existing software for those who would like to get involved with such things.

Visitors are welcome to attend.

10-Oct-80 10:45:35-MDT,00000511;000000000001
Date: 10 Oct 1980 1045-MDT
From: KELLER
Subject: Multiprocessor simulator
To: AMPS:

<amps> now contains a multiprocessor version of the amps simulator. This is a modification of the former amps simulator, and is "upward compatible" except for needing a fix in the storage reclaimer. The relevant files are rose.exe, rose.pas(code), rose.doc(documentation). Experimentation is welcome. The documentation describes parameters that can be changed to get different load distribution effects.

10-Oct-80 10:55:56-MDT,0000000356;000000000001
Date: 10 Oct 1980 1055-MDT

From: KELLER
Subject: References
To: AMPS:

I've started a list of "new" references that are of likely interest to our group. In particular, reviewing one of these papers would be a nice way to contribute to the weekly meetings. The list is on <amps>newrefs.mss. Please feel free to add entries of your own.

16-Oct-80 13:34:39-MDT,00000000494;000000000003
Date: 16 Oct 1980 1334-MDT
From: KELLER
Subject: FGL tutorial
To: AMPS:
cc: bboard

As previously announced, Friday 10/17 at 11 am in 3141 will be the first in a series of tutorials on FGL/AMPS. This meeting will cover rudimentary aspects of FGL (Function Graph Language), including graphical and textual representations.

The next "regular" AMPS meeting will be the following Friday, when P.A. Subrahmanyam will discuss POPART: a Prcducer of Parsers and Related Tools.

22-Oct-80 13:05:33-MDT,00000934;000000000003
Date: 22 Oct 1980 1305-MDT
From: KELLER
Subject: Meeting Friday
To: AMPS:
cc: bboard

The next AMPS meeting will be this Friday, Oct. 24. The following describes the scheduled presentation:

POPART -- a Producer of Parsers and Related Tools
by P.A. Subrahmanyam

POPART nominally produces parsers tailored to Interlisp conventions for grammars specified in a BNF variant that permits regular expressions. POPART also generates a set of tools for each grammar, including: a pattern matcher and replacer, a "structured" editor, a "pretty printer", and a "development system".

Time permitting, the speaker will mention/elaborate upon extensions/uses/... for POPART in some ongoing/forthcoming projects at Utah.

Load average permitting, he will attempt a brief demo. No promises.

The next "tutorial" (FGL part 2) will be next Friday.

31-Oct-80 14:06:55-MST,00000000326;000000000001
Date: 31 Oct 1980 1406-MST
From: LINDSTROM
Subject: FSTREAM example
To: lepreau, hudak, tanaka, YAMAZAKI
cc: keller

As a follow-up to the AMPS tutorial today, I have prepared a program showing the construction of a pseudo random number stream. Consult <LINDSTROM>FSTREAM.*

for details. Thanks for coming!

 5-Nov-80 20:36:19-MST,00000000308;000000000003
Date: 5 Nov 1980 2036-MST
From: LINDSTROM
Subject: Yamazaki talk at AMPS meeting
To: AMPS: ;
cc: bboard

The next regular meeting of the AMPS group will take place Fri. 11/7 at 11am
in 3141. Takemi Yamazaki will discuss the concepts and design of the MICS
muitiprocessor computer system.
----
k, tanaka, YAMAZAKI
cc: keller

As a follow-up to the AMPS tutorial today, I have prepared a program showing
the construction of a pseudo random number stream. Consult <LINDSTROM>FSTREAM.*
for details. Thanks for coming!

****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****
USER: LEPREAU FILE NAME: AMPS_MAIL.TXT.1 ACCOUNT: OVERHEAD PAGES: 000007
****